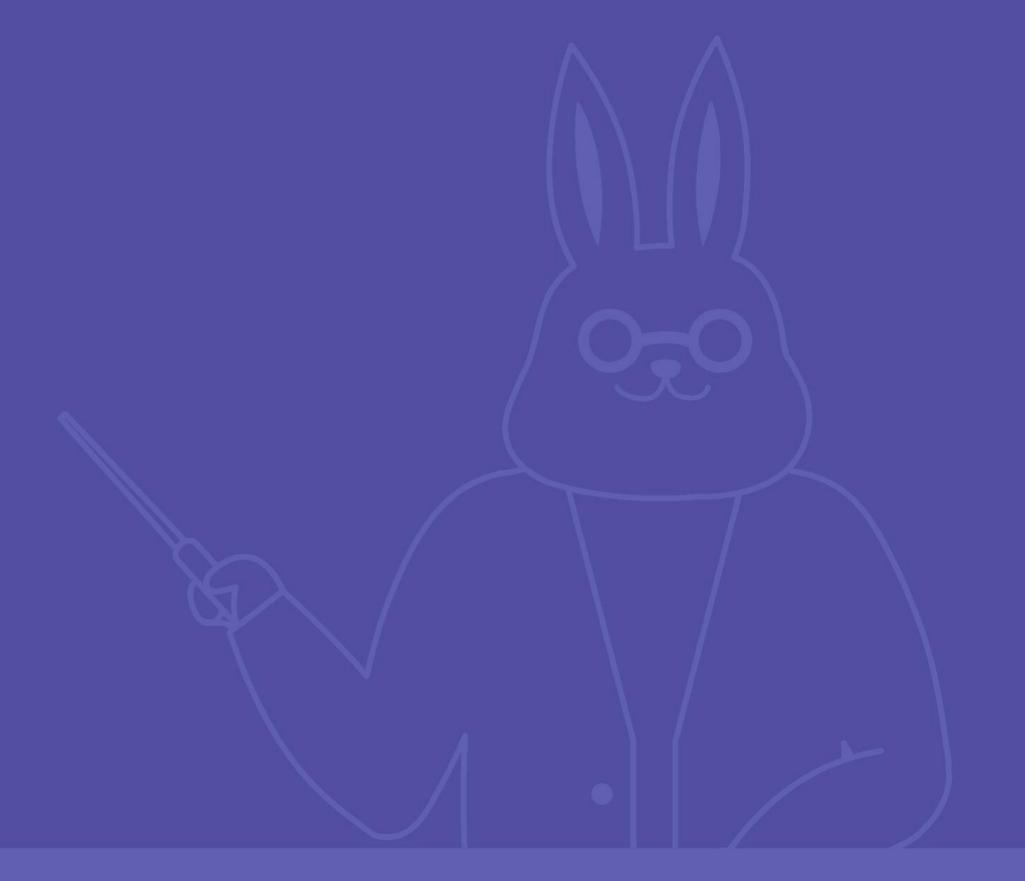


MongoDB 기초

O1 MongoDB 개요





- 01. MongoDB 체험하기
- 02. 데이터베이스계의 이단아 NoSQL
- 03. MongoDB 소개 및 분산 컴퓨팅
- 04. JS 친화적 MongoDB
- 05. MongoDB 활용



두괄식 설명

맥락 이해가 쉽도록 두괄식으로 설명한다

예시와 비유

쉬운 이해를 위해 이론 부분은 예시와 비유를 적극 활용한다

실습 중심

해보지 않으면 이해할 수 없으므로 배우고 난 후 바로 실습해본다



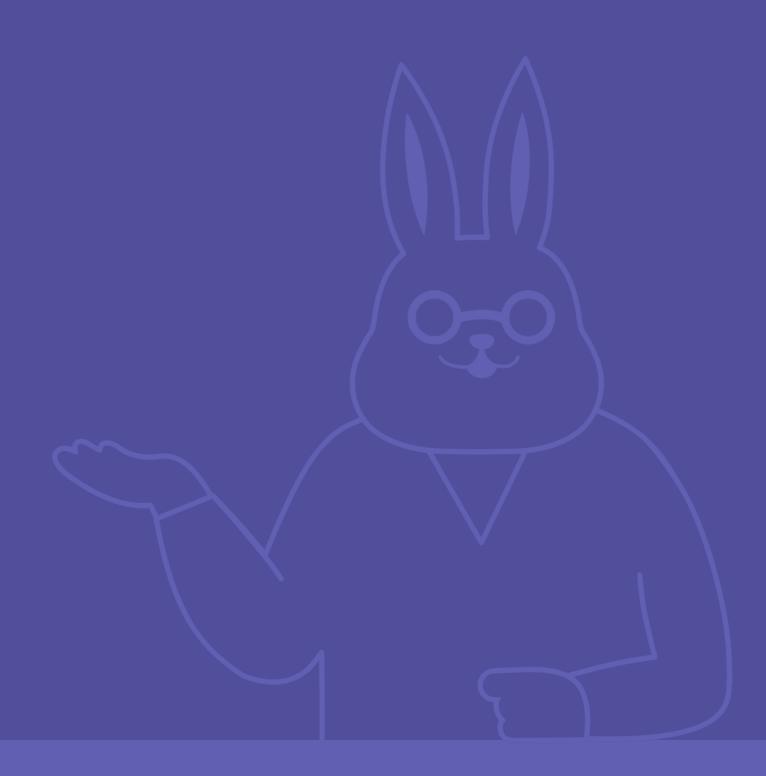
개발 지식 학습의 3가지 단계

- 1. 어떤 지식이 있는지 몰라 찾아보지도 못하는 상태
- 2. 이론적 뼈대를 알고 있어, 스스로 검색하여 학습할 수 있는 단계
- 3. 유사한 기술을 많이 경험하여 스스로 추론할 수 있는 단계

본 강좌는 여러분이 2단계가 되도록 만드는 것이 목표

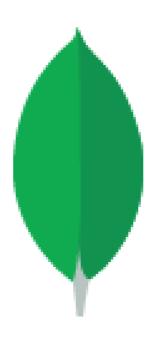
01

MongoDB 체험하기



Confidential all rights reserved

♥ 첫번째로 해볼 작업



우선 MongoDB에 대해서 느껴보자!



☑ DB 접속하기



MongoDB Shell

> use myDatabase

❷ 정보 저장하기



MongoDB Shell

- > use myDatabase
- > db.initialCollection.insertOne({hello: "world"})

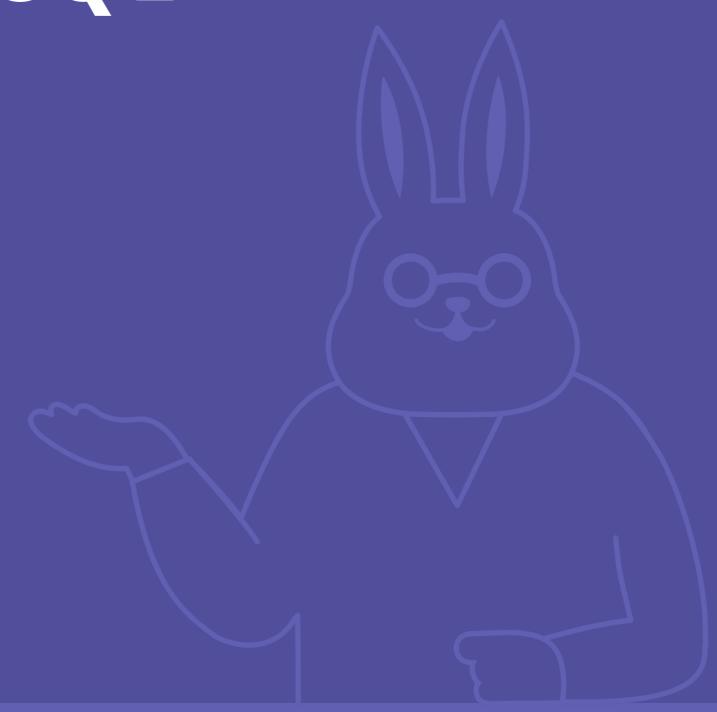
❷ 정보 확인하기



MongoDB Shell

- > use myDatabase
- > db.initialCollection.insertOne({hello: "world"})
- > db.initialCollection.find()

데이터베이스계의 이단아 NoSQL



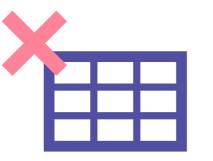
❷ 이전 체험에서 느낀 점



이전 체험에서 어떤 점을 느낀점은?



질의 명령어가 SQL이 아니다



정보의 형식을 미리 정하지 않는다

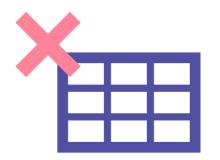
ObjectId("...")

_id 값이 자동 저장되었다 ❷ 이전 체험에서 느낀 점

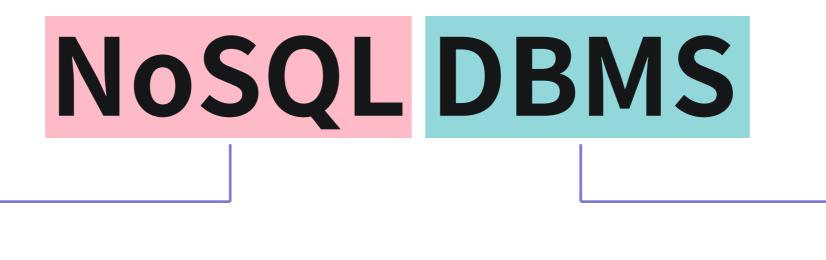
NoSQL의 특징을 담고 있는 MongoDB



질의 명령어가 SQL이 아니다



정보의 형식을 미리 정하지 않는다



Not Only SQL

전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 제공하는 DB

Database Management System

데이터를 저장하고 관리하는 시스템 실생활에 간략하게 DB라고 부른다

✓ NoSQL DBMS의 유행 배경

대표적인 관계형 데이터베이스 제작시기: 1990년대 이전



MySQL (1995년 출범)



PostgreSQL (1996년 출범)



Oracle DB (1979년 출범)

☑ NoSQL DBMS의 유행 배경

대표적 NoSQL DB 제작시기: 2009년 이후



MongoDB (2009년 출범)



ElasticSearch (2010년 출범)



Redis (2009년 출범)

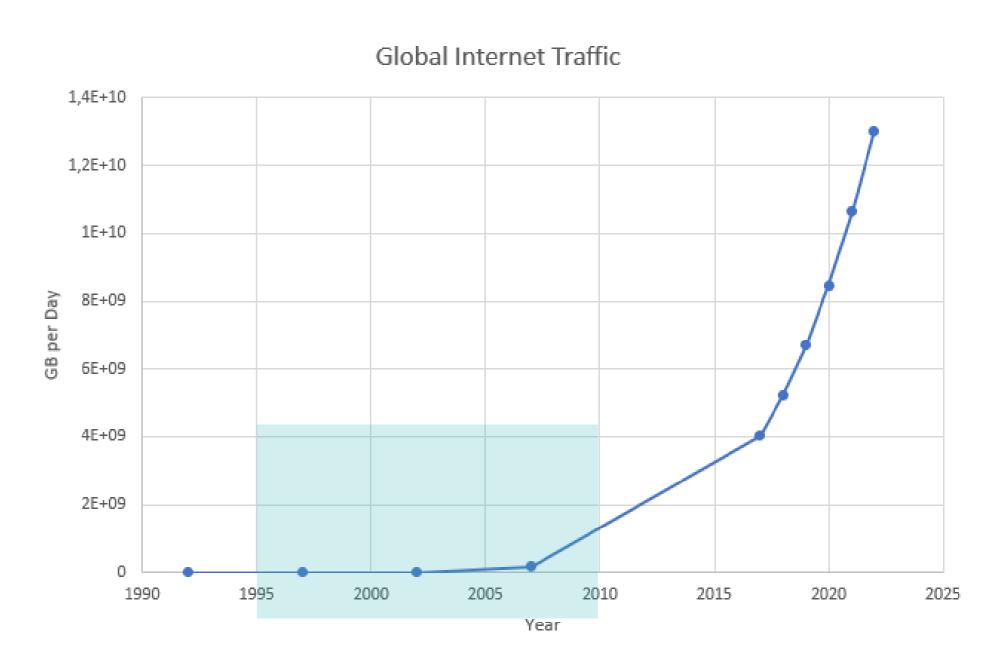
1990년대와 2009년 사이에 무슨 일이? RDBMS NoSQL

✓ NoSQL DBMS의 유행 배경

1990~2010년 사이에 무슨 일이?



아이폰의 등장 (2007년)

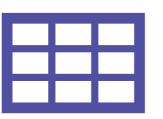


인터넷의 확산

☑ NoSQL DBMS의 유행 배경

전통적 RDBMS의 규칙







ACID 원칙을 준수

2차원의 테이블 형태

SQL을 통한 질의

☑ NoSQL DBMS의 유행 배경

서버에 요구되는 정보 처리량의 급속한 증가 정보 처리량을 늘리려면 규칙을 깨야 한다













안정성에 중점을 둔 RDBMS

확장성과 성능 최적화에 특화된 NoSQL

특정 기능에 대해서는 RDBMS의 규칙이 비효율적이다













범용적으로 활용 가능한 RDBMS

각각의 기능에 특화된 NoSQL

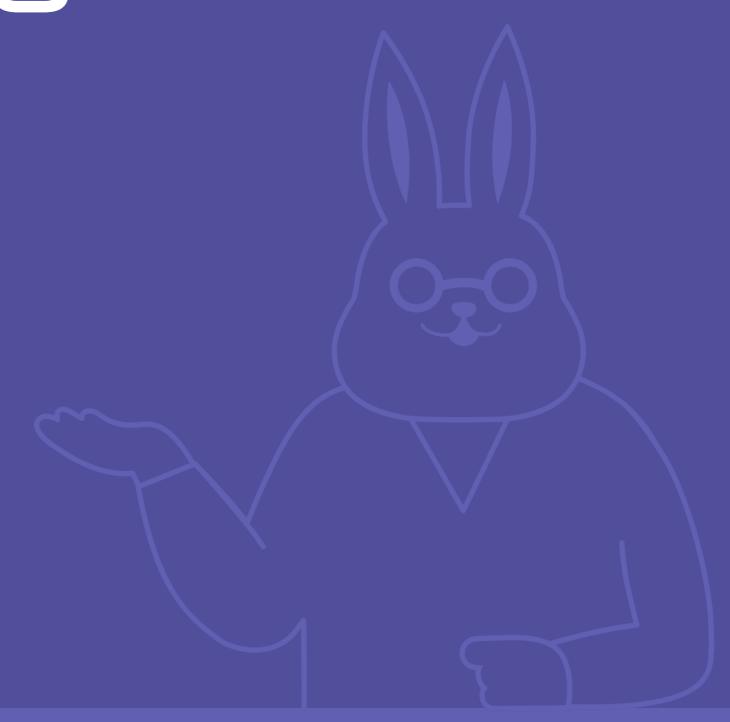
NoSQL DBMS

Not Only SQL

특징

전통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 제공하는 DB 기존 RDBMS가 일관성 모델 때문에 가질 수 없었던 확장성, 유연성, 고성능, 고기능성을 확보함

MongoDB 소개 및 분산 컴퓨팅



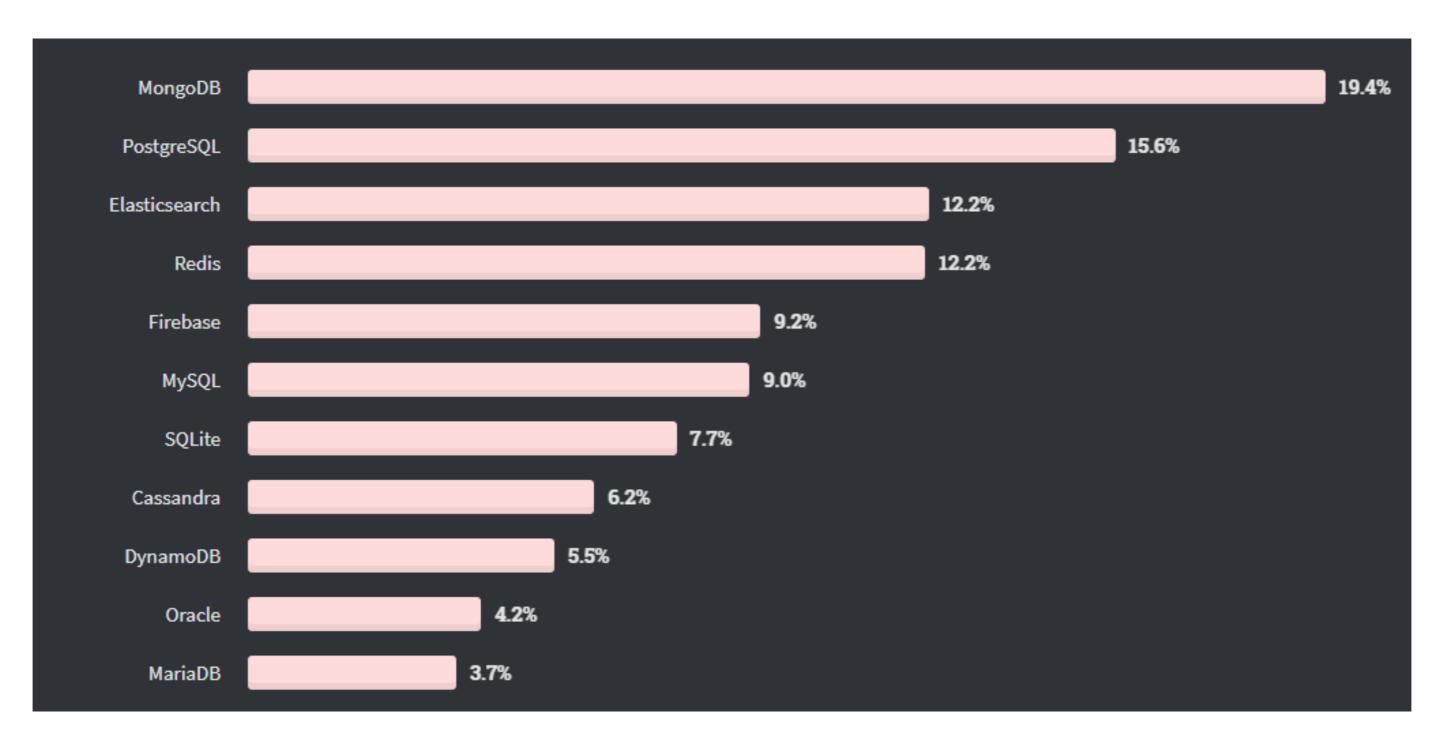
Confidential all rights reserved

❷ MongoDB의 인기

MongoDB는 가장 유명한 NoSQL DBMS이다

☑ MongoDB의 인기

StackOverflow 설문: 가장 배우고 싶은 DBMS 기술은?



3년 연속 가장 배우고 싶은 NoSQL DBMS

❷ MongoDB의 사용처



PC방 관리 시스템에 도입



알림 서버에 도입





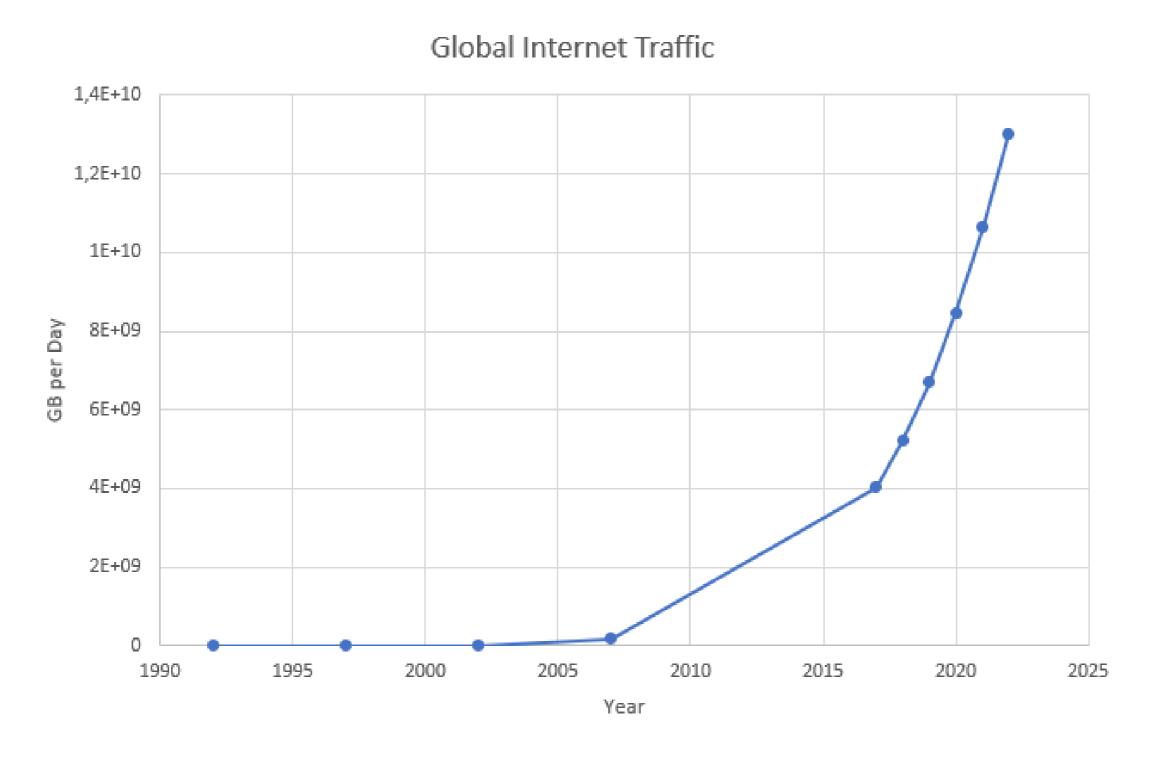
그 외 각종 서비스..

☑ MongoDB의 인기

왜 MongoDB는 배우고 싶은 기술일까?

현대적 설계로 고성능을 내면서 동시에

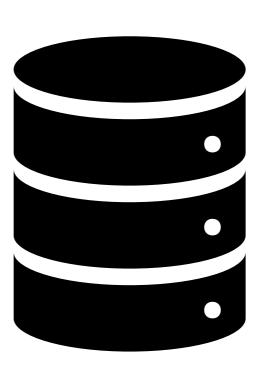
많은 기능과 안정화가 이루어진 다목적 DB



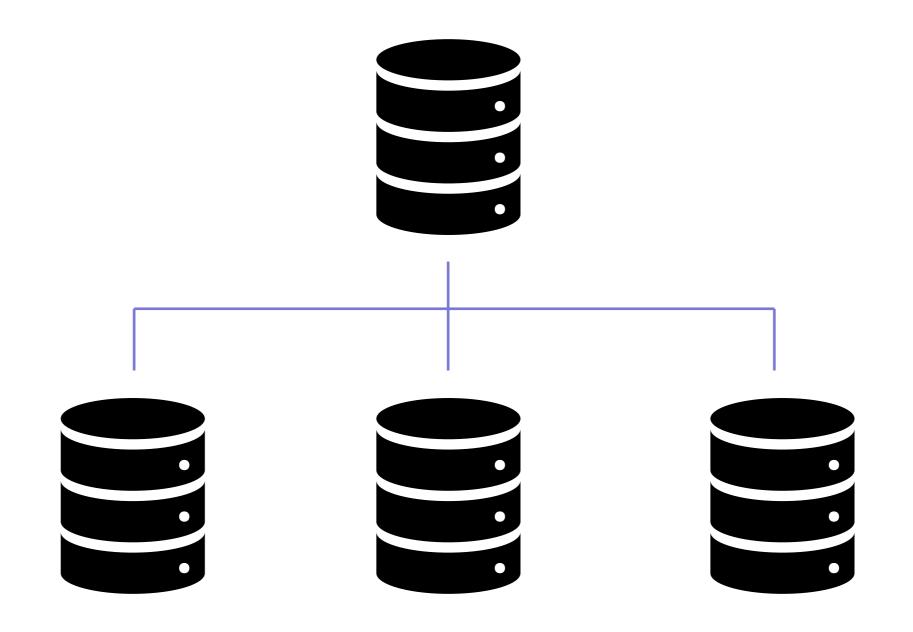
인터넷의 부상으로 서버의 요구 처리량이 증가



카카오톡을 예시로 들어보자면 하루 평균 송·수신 메시지 110억건 초당 평균 13만건의 송·수신



과거에는 하나의 DBMS로 처리하는 상황이 빈번했다 기존 RDBMS는 하나의 인스턴스에서 작동하는 것이 기본값

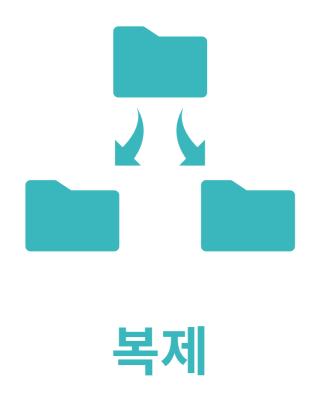


요즘은 처리할 데이터양이 많아서 **분산 컴퓨팅**이 빈번하다 현대적 DBMS는 **분산 컴퓨팅**을 하는 것이 기본값

MongoDB에서 지원하는 분산 컴퓨팅은 무엇이 있을까?



❷ 분산 컴퓨팅의 방식



복사하여 저장하는 방식

안정성을 높이기 위한 방식 원본 서버가 망가져도 정상 서비스 가능



샤딩

나누어 저장하는 방식

성능을 향상하기 위한 방식 읽기, 쓰기 성능 향상 가능

▼ 분산 컴퓨팅 관련 MongoDB 주의점

단일 인스턴스 DBMS에서 만약 쓰기 작업을 하다가 서버가 갑자기 꺼진다면?



데이터가 의도치 않게 중간 상태로 저장될 수 있다

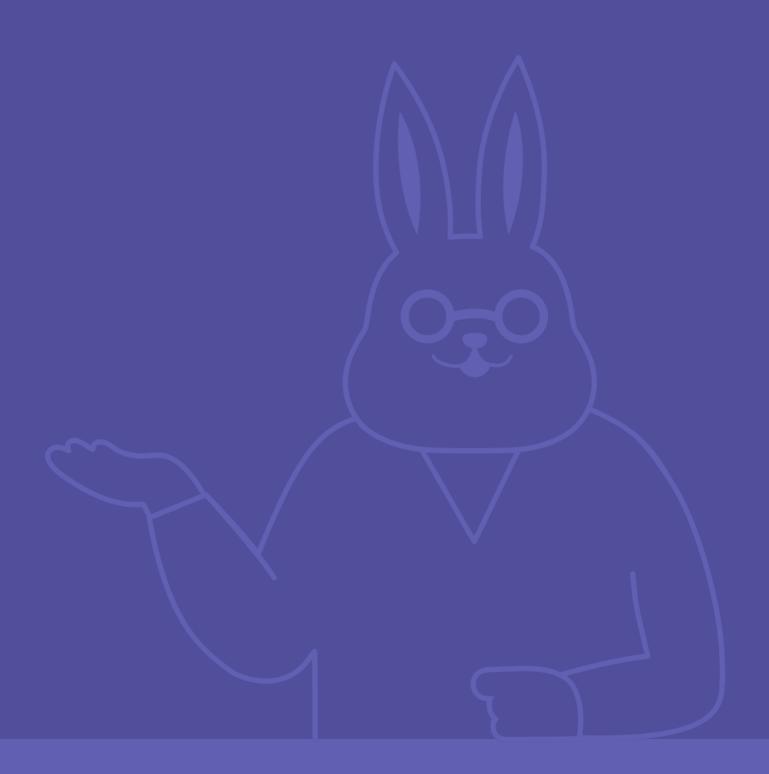


데이터가 항상 저장 완료하거나 저장 실패한 상태로만 존재

WHY?

MongoDB는 분산 컴퓨팅을 하는 것을 가정하고 쓰기 기본 설정이 되어 있다! Write-Concern, Read-Concern 설정을 알아야 제대로 사용 가능하다!

JS 친화적 MongoDB



Confidential all rights reserved

☑ 고급 기능 내장의 문제점

분산 컴퓨팅 기본값 설정의 문제점

입문이 너무 어렵다!!

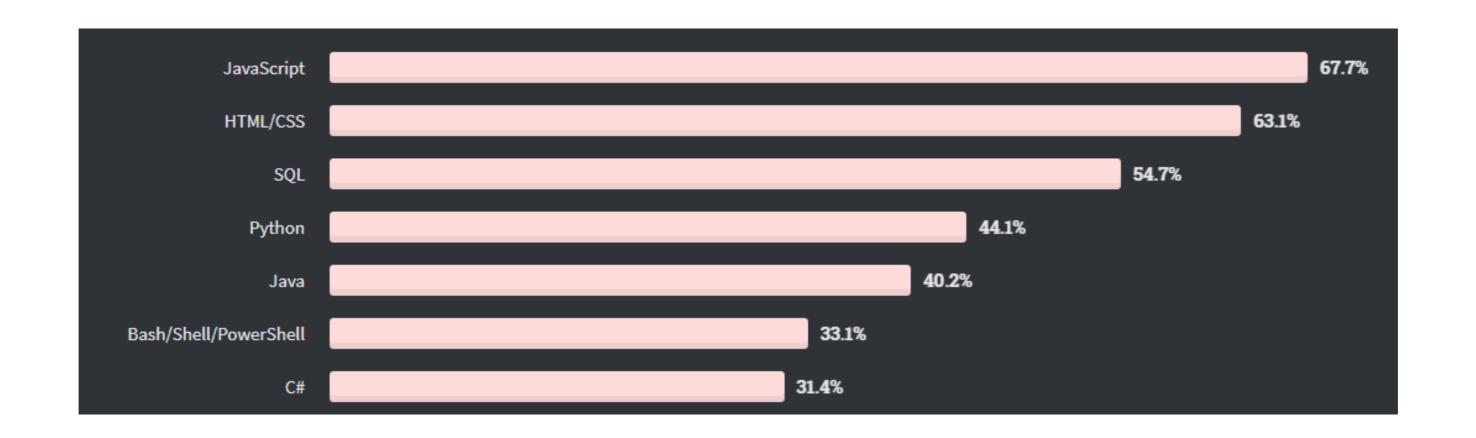
처음 배워야할 개념들

쓰기 설정?

샤딩?
복제?
ACID?
분산 컴퓨팅?

❷ 입문자를 위한 JS 채택

개발자 중 가장 많은 개발자는 웹 개발자



MongoDB 탄생 당시 신생 JS 엔진이 등장!

❷ 입문자를 위한 JS 채택

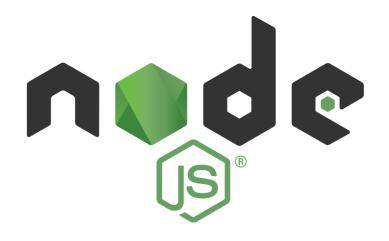


2008년 JS 속도를 획기적으로 향상시킨 V8 엔진 등장





2008년 Chrome의 부상



2009년 백엔드 개발이 가능한 NodeJS 등장

☑ 입문자를 위한 JS 채택

V8엔진의 성능향상으로 인한 웹 개발자의 대규모 업종전환 MongoDB도 이런 대세를 활용하여 SQL을 몰라도 개발 가능한

V8엔진의 JS 기반 쉬운 DBMS를 표방함

● 내부 명령어가 JS로 구성되어 있음

MongoDB Shell

- > use myDatabase
- > db.initialCollection.insertOne({hello: "world"})
- > db.initialCollection.find()

04 JS 친화적 MongoDB

☑ JSON과 비슷한 BSON 자료구조

```
BSON 구조 예시
                         field: value
 name: "sue",
                         field: value
 age: 26,
                          ——— field: value
 status: "A",
 groups: ["news", "sports"] — field: value
```

JSON과 유사한 BSON 구조로 정보를 저장

Javascript Object Notation

☑ 스키마가 없는 방식

BSON 구조 예시

```
name: "sue",
age: 26,
status: "A",
groups: ["news", "sports"]
name: "jim",
color: "white",
cloth: { color: "white", size: 150 }
```

테이블 구조 예시

name	age
sue	26
jim	30

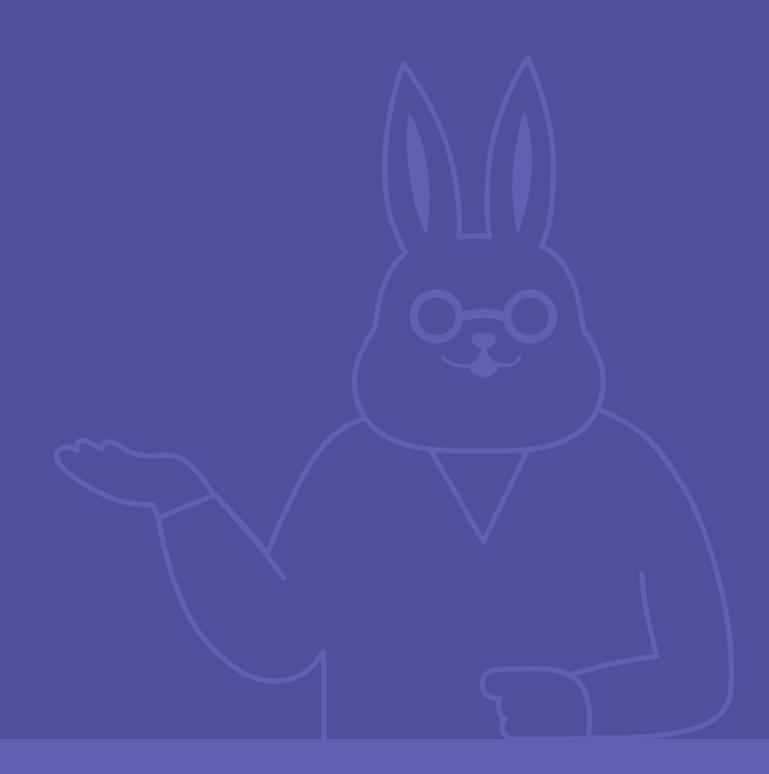


MongoDB가 JS를 사용해서 얻은 특징

- 1. 웹 개발자에게 쉬운 입문이 가능하다
- 2. BSON 자료형 사용
- 3. 내부 명령어를 JS 형식으로 사용

05

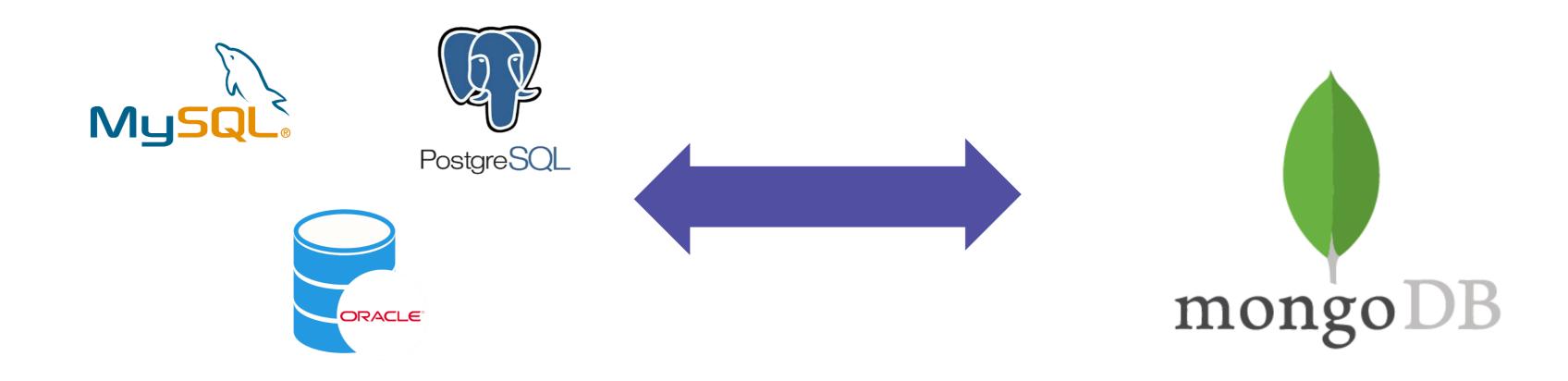
MongoDB활용



Confidential all rights reserved

❷ MongoDB가 대체 가능한 기능

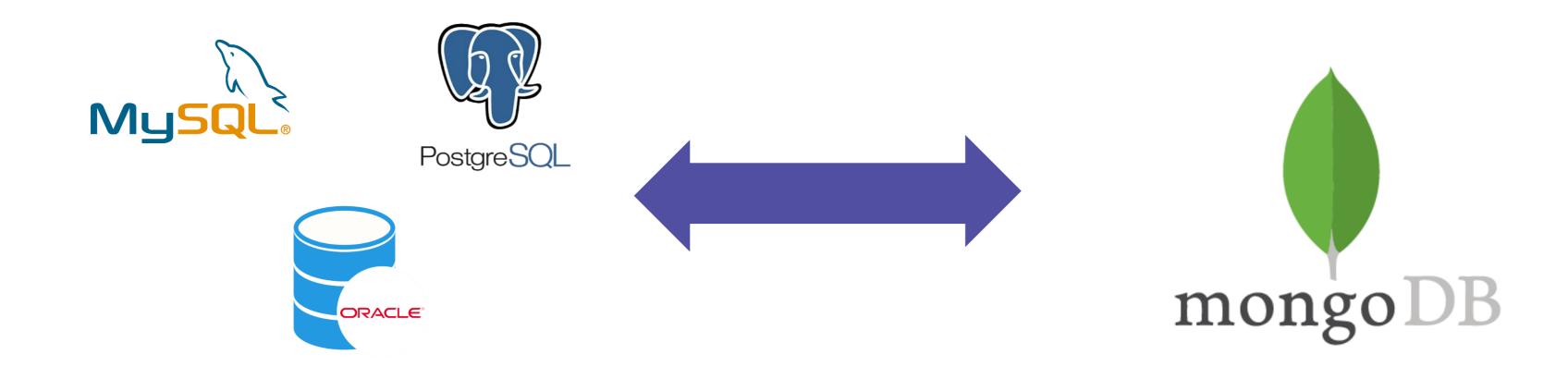
MongoDB는 범용적인 DBMS로 활용 가능하다



단, 일반적인 상황에서 활용하기에는 무리가 있다

05 MongoDB 활용

❷ MongoDB가 대체 가능한 기능



적어도 25년간은 유지보수된 기술

다양한 개발자와 관련 문서

개발된지 10년 된 기술

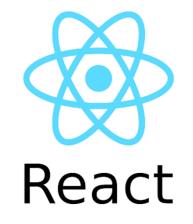
이제 어느정도 안정성이 갖춰짐

❷ MongoDB의 특장점



- 1. JS에 친화적이다
- 2. 성능 확장이 쉽다
- 3. 높은 성능을 낼 수 있다
- 4. 유연한 구조로 저장할 수 있다
- 5. 다양한 자료형을 지원한다

☑ JS 기반 프로젝트



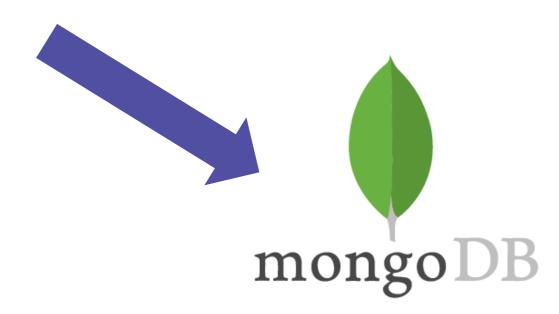




프로젝트 언어를 JS로 통일시키고자 할 때

정보 변환 없이 전달 가능

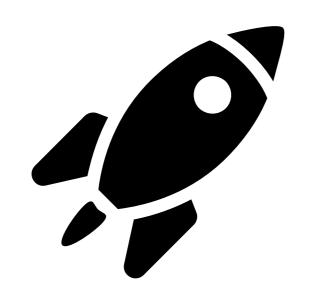




기존의 JSON 방식의 DB에서 좀 더 많은 기능과 성능이 필요할 때

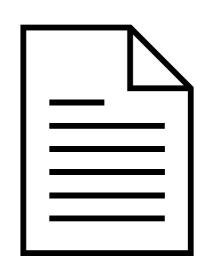
입문용 DB에서 실전용 DB로 전환

☑ 저장할 정보의 형태가 자주 변경되는 경우



파일럿 프로젝트

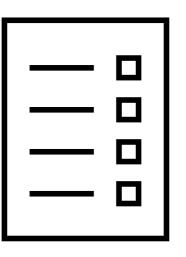
앞으로 몇 명이 이용할지 예측이 안되는 프로젝트 한참 개발 중이고 상황에 따라 데이터 구조가 쉽게 바뀔 수 있는 신규 프로젝트



비 정형화된 정보를 전산화

각종 예외사항이 적혀 있는 수기로 작성된 과거의 주문 정보 로그 데이터 형식의 정보를 저장할 때 정보를 정형화하기에 어려운 경우

❷ 높은 성능이 필요할 경우



안정성보다 높은 성능이 필요한 경우

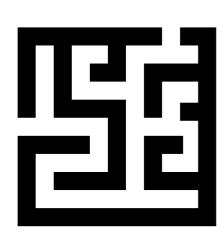
NoSQL이 기존 DBMS보다 수십배는 더 높은 성능을 가짐 안정성보다 높은 성능이 필요할 때 ex) 카카오 모빌리티의 지도정보, SNS 정보 저장

☑ 그렇다면 활용하면 안되는 경우는 언제 일까?



높은 성능보다 안정성, 무결성이 중요한 경우

결제 시스템, 예약 시스템 데이터 무결성이 중요한 경우



복잡한 쿼리가 빈번한 경우

JOIN, UNION과 같은 연산을 처리할 때 좀 더 복잡해진다 RDB와 비교했을 때 없는 함수가 꽤 있다

크레딧

/* elice */

코스 매니저 이재성

콘텐츠 제작자 정승호

강사 정승호

감수자 정승호

디자이너 강혜정

연락처

TEL

070-4633-2015

WEB

https://elice.io

E-MAIL

contact@elice.io

